Chapter VII

A Layered Model for Building Ontology Translation Systems

Oscar Corcho, Intelligent Software Components, Spain*

Asunción Gómez-Pérez, Universidad de Madrid, Spain

Abstract

In this chapter we present a model for building ontology translation systems between ontology languages and/or ontology tools, where translation decisions are defined at four different layers: lexical, syntax, semantic, and pragmatic. This layered approach provides a major contribution to the current state of the art in ontology translation, since it makes ontology translation systems easier to build and understand and, consequently, to maintain and reuse. As part of this model, we propose a method that guides in the process of developing ontology translation systems according to this approach. The method identifies four main activities: feasibility study, analysis of source, and target formats, design, and implementation of the translation system, with their decomposition in tasks, and recommends the techniques to be used inside each of them.
Introduction

An ontology is defined as a “formal explicit specification of a shared conceptualization” (Studer et al., 1998); that is, an ontology must be machine readable (it is formal), all its components must be described clearly (it is explicit), it describes an abstract model of a domain (it is a conceptualization), and it is the product of a consensus (it is shared).

Ontologies can be implemented in varied ontology languages, which are usually divided in two groups: classical and ontology markup languages. Among the classical languages used for ontology construction, we can cite (in alphabetical order): CycL (Lenat & Guha, 1990), FLogic (Kifer et al., 1995), KIF (Genesereth & Fikes, 1992), LOOM (MacGregor, 1991), OCML (Motta, 1999), and Ontolingua (Gruber, 1992). Among the ontology markup languages used in the context of the Semantic Web, we can cite (in alphabetical order): DAML+OIL (Horrocks & van Harmelen, 2001), OIL (Horrocks et al., 2000), OWL (Dean & Schreiber, 2004), RDF (Lassila & Swick, 1999), RDF Schema (Brickley & Guha, 2004), SHOE (Luke & Hefflin, 2000), and XOL (Karp et al., 1999). Each of these languages has its own syntax, its own expressiveness, and its own reasoning capabilities provided by different inference engines. Languages also are based on different knowledge representation paradigms and combinations of them (frames, first order logic, description logic, semantic networks, topic maps, conceptual graphs, etc.).

A similar situation applies to ontology tools: several ontology editors and ontology management systems can be used to develop ontologies. Among them, we can cite (in alphabetical order): KAON (Maedche et al., 2003), OilEd (Bechhoefer et al., 2001), OntoEdit (Sure et al., 2002), the Ontolingua Server (Farquhar et al., 1997), OntoSaurus (Swartout et al., 1997), Protégé-2000 (Noy et al., 2000), WebODE (Arpírez et al., 2003), and WebOnto (Domingue, 1998). As in the case of languages, the knowledge models underlying these tools have their own expressiveness and reasoning capabilities, since they are also based on different knowledge representation paradigms and combinations of them. Besides, ontology tools usually export ontologies to one or several ontology languages and import ontologies coded in different ontology languages.

There are important connections and implications between the knowledge modeling components used to build an ontology in such languages and tools, and the knowledge representation paradigms used to represent formally such components. With frames and first order logic, the knowledge components commonly used to build ontologies are (Gruber, 1993) classes, relations, functions, formal axioms, and instances; with description logics, they are usually (Baader et al., 2003) concepts, roles, and individuals; with semantic networks, they are: nodes and arcs between nodes; etc.
A Tool for Working with Web Ontologies
www.igi-global.com/article/tool-working-web-ontologies/2804?camid=4v1a